Cambridge International AS & A Level

Cambridge International Examinations
Cambridge International Advanced Subsidiary and Advanced Level

COMPUTER SCIENCE **9608/11**

Paper 1 Written Paper **May/June 2016**

MARK SCHEME

Maximum Mark: 75

Published

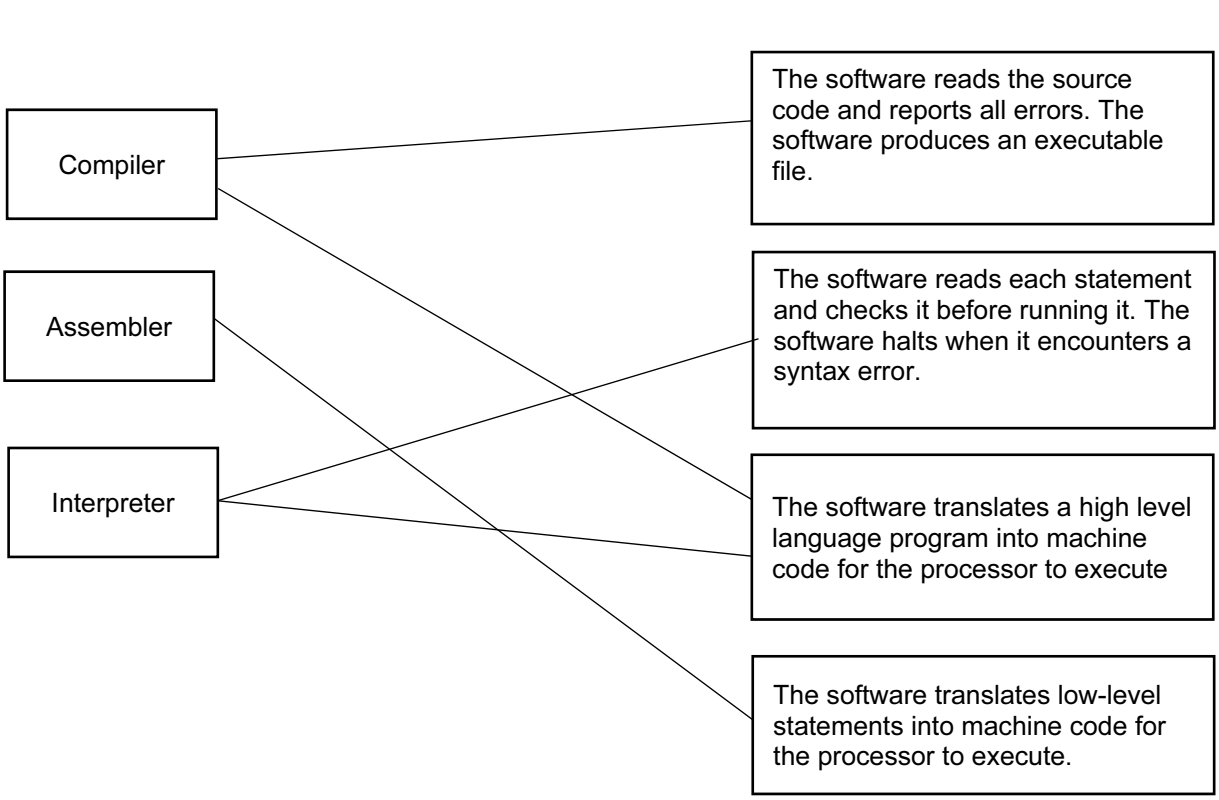
This mark scheme is published as an aid to teachers and candidates, to indicate the requirements of the examination. It shows the basis on which Examiners were instructed to award marks. It does not indicate the details of the discussions that took place at an Examiners' meeting before marking began, which would have considered the acceptability of alternative answers.

Mark schemes should be read in conjunction with the question paper and the Principal Examiner Report for Teachers.

Cambridge will not enter into discussions about these mark schemes.

Cambridge is publishing the mark schemes for the May/June 2016 series for most Cambridge IGCSE®, Cambridge International A and AS Level components and some Cambridge O Level components.

1 **One mark** for each box on the left. **[3]**



Marks allocated as follows:

Compiler – **1 mark** for **two** correct connecting lines
Assembler – **1 mark** for **one** correct connecting line
Interpreter – **1 mark** for **two** correct connecting lines

2 **(a)** 00110111 **[1]**

(b) 83 **[1]**

(c) 10011010 **[2]**

Marks allocated as follows:

1 mark for the most significant bit
1 mark for the remaining 7 bits

(d) 78 **[1]**

3 (a) **Four** points from: **[4]**

- The Program Counter (PC) holds the address of the next instruction to be fetched
- The address in the Program Counter (PC) is copied to the Memory Address Register (MAR)
- The Program Counter (PC) is incremented
- The instruction is copied to the Memory Data Register (MDR)
  - …. from the address held in the Memory Address Register (MAR)
- The instruction from the Memory Data Register (MDR) is copied to the Current Instruction Register (CIR)

(b) **One mark** for each statement or letter in the correct place. **[4]**

At the end of the cycle for the current instruction **B**
If the interrupt flag is set, **D**, **A** and **C**
The interrupted program continues its execution

At the end of the cycle for the current instruction the processor checks if there is an interrupt.
If the interrupt flag is set, the register contents are saved, the address of the Interrupt Service Routine (ISR) is loaded to the Program Counter (PC) and when the ISR completes, the processor restores the register contents.

The interrupted program continues its execution.

4 (a) **Three** from: **[3]**

- The height/amplitude of the (sound) wave is determined.
- At set (time) intervals // by example of sensible time period.
- To get an approximation of the sound wave
- And encoded as a sequence of binary numbers // and converted to a digital signal.
- Increasing the sampling rate will improve the accuracy of the recording.

(b) (i) **No mark** awarded for identifying method. **Three marks** for justification. **[3]**

Lossy – Three points from:
- The human ear will not notice that the decompressed stream will not be identical to the original (file) / that parts of the original data have been discarded / removed / deleted.
- File size reduction is greater than using lossless.
- Email has limits on file sizes (on attachments) / a smaller file will take less time to transmit.
- The file may not need to be of high precision / accuracy.
- The producer has requested an mp3 file.

Lossless – Three points from:
- The file needs to be high precision / accuracy.
- None of the original data is lost / the decompressed file will be identical to the original.
- The producer has requested a flac file.

(ii) **Three** points from: **[3]**

- Lossless method of compression.
- Reduces (the physical size of) a string of adjacent, identical characters/pixels / bytes etc..
- The repeating string (a run) is encoded into two values.
- One value represents the number of (identical) characters in the run (the run count).
- The other value is the code of the character / colour code of pixel etc. in the run (the run value).
- The run value and run count combination may be preceded by a control character.
- Any valid example given.

(iii) **Two marks** for **three** correct rows, **one mark** for **two** correct rows. **[2]**

Row 1: 153 10 255 3 153 3
Row 2: 153 9 255 6 153 1
Row 3: 153 7 255 9

Alternative correct answer:

Row 1: 153 9 255 2 153 2
Row 2: 153 8 255 5 153 0
Row 3: 153 6 255 8

5 **One mark** per row. **[4]**

No mark if more than **one tick** in any row.

Description	Open source	Shareware	Commercial
Software is purchased before it can be used			✓
Source code comes with the software	✓		
Software is provided free on a trial basis		✓	
The software can be modified by the user	✓		

6 **One mark** for identifying the principle, **one mark** for an example that is in the context of this scenario.

Maximum of two marks per principle. **Maximum of three** principles. **[6]**

- PUBLIC / Software engineers shall act consistently with the public interest.
  - Example in context
- CLIENT AND EMPLOYER / Software engineers shall act in a manner that is in the best interests of their client and employer (consistent with the public interest.)
  - Example in context
- PRODUCT / Software engineers shall ensure that their products and related modifications meet the highest professional standards possible.
  - Example in context
- JUDGEMENT / Software engineers shall maintain integrity and independence in their professional judgment.
  - Example in context
- MANAGEMENT / Software engineering managers and leaders shall subscribe to and promote an ethical approach to the management of software development and maintenance.
  - Example in context
- PROFESSION / Software engineers shall advance the integrity and reputation of the profession (consistent with the public interest).
  - Example in context
- COLLEAGUES / Software engineers shall be fair to and supportive of their colleagues.
  - Example in context
- SELF / Software engineers shall participate in lifelong learning regarding the practice of their profession and shall promote an ethical approach to the practice of the profession.
  - Example in context

7 (a) Internet Protocol **[1]**

(b) **[4]**

Address	Denary / Hexadecimal	Valid or Invalid	Reason
3.2A.6AA.BBBB	Hexadecimal	Invalid	**One** point from: • This is more than 32 bits • 6AA /BBBB in Hex is bigger than FF / 255 in denary • 6AA / BBBB uses more than 8 bits / a byte • The third / fourth group is bigger than FF / 255 in denary • The third / fourth group uses more than 8 bits / a byte
2.0.255.1	Denary	Valid	There are 4 bytes, each 255 or below // All the values are in the range 0 - 255
6.0.257.6	Denary	Invalid	257 is above 255 // The third group is above 255
0A.78.F4.J8	Hexadecimal	Invalid	J is not a valid hexadecimal digit // J8 is not a valid Hex number

One mark for each combination of valid or invalid **and** the reason.

(c) Two points from: **[2]**

- Public address can be reached across the Internet.
- Private address can only be reached internally/through the LAN/Intranet // private address cannot be reached across the Internet.
- NAT (Network Address Translation) is necessary for a private IP address to access the Internet directly.
- A private address is more secure than a public address // A public address is less secure than a private address.
- Public addresses are provided by ISP / assigned by InterNIC // Private addresses are assigned by the router (of the network concerned).
- Public addresses are unique (to the Internet) // Private addresses (are unique within their network, but) can be duplicated within other (discrete) networks.
- 10.0.0.1 to 10.255.255.254 and 172.16.0.1 to 172.31.255.254 and 192.168.0.1.to 192.168.255.254 form the private address space // IP addresses from the private address space are never assigned as public.

8 (a) (i) Database Management System **[1]**

(ii) **One mark** for identifying the way in which the data security is ensured, and **one mark** for a further description.

Maximum of two marks per method. **Maximum of two** methods. **[4]**

- Issue usernames and passwords...
  - stops unauthorised access to the data
  - any further expansion e.g. strong passwords / passwords should be changed regularly etc…
- Access rights / privileges...
  - so that only relevant staff / certain usernames can read/edit certain parts of the data
  - can be read only, or full access / read, write and delete
  - any relevant example e.g. only class tutors can edit details of pupils in their tutor group
- Create (regular / scheduled) backups...
  - in case of loss/damage to the live data a copy is available
  - any relevant example e.g. backing up the attendance registers at the end of each day and storing the data off-site/to a separate device
- Encryption of data...
  - if there is unauthorised access to the data it cannot be understood // needs a decryption key
  - any relevant example e.g. personal details of pupils are encrypted before being sent over the Internet to examination boards
- Definition of different views...
  - composed of one or more tables
  - controls the scope of the data accessible to authorised users
  - any relevant example e.g. teachers can only see their classes
- Usage monitoring / logging of activity...
  - creation of an audit /activity log
  - records the use of the data in the database / records operations performed by all users / all access to the data
  - any relevant example, e.g. Track who changed a student's grade

(iii) **Two points** from: **[2]**

- Set up search criteria
- To find / retrieve / return the data that matches the criteria
- Any relevant example e.g. find pupils who were absent on a particular day

(iv) **Three points** from: **[3]**

- By storing data in (separate) linked tables data redundancy is reduced / data duplication is controlled...
- Compatibility / data integrity issues are reduced as data only needs to be updated once / is only stored once.
- Unwanted or accidental deletion of linked data is prevented as the DBMS will flag an error.
- Program - data dependence is overcome.
- Changes made to the structure of the data have little effect on existing programs.
- Ad-hoc / complex queries can be more easily made as the DBMS will have a query language/ QBE form.
- Unproductive maintenance is eliminated as changes only need to be made once (rather than changing multiple programs).
- Fields can be added or removed without any effect on existing programs (that do not use these fields).
- Security / privacy of the data is improved as each application only has access to the fields it needs.
- There is better control of data integrity as the DBMS (uses its Data Dictionary) to perform validation checks on data entered.

(b) **(i)** **Two points** from: **[2]**

- The Primary Key in **CLASS** is ClassID
- The Foreign Key of **CLASS-GROUP** is ClassID.
- The Primary Key of CLASS is also included in CLASS-GROUP as a Foreign Key, (which links to CLASS table)

(ii) Many-to-one **[1]**

(iii) **One mark** per statement. Several statements may be on the same line. **[4]**

```
SELECT StudentID, FirstName
FROM STUDENT
WHERE TutorGroup = "10B" // WHERE (TutorGroup = "10B")
ORDER BY LastName ASC;
```

(iv) **One mark** per statement. Several statements may be on the same line. **[4]**

```
SELECT STUDENT.LastName
FROM STUDENT, CLASS-GROUP
WHERE ClassID = "CS1" // WHERE (ClassID = "CS1")
AND CLASS-GROUP.StudentID = STUDENT.StudentID;
```

One mark per statement. Several statements may be on the same line.

```
SELECT STUDENT.LastName
FROM STUDENT INNER JOIN CLASS-GROUP
ON CLASS-GROUP.StudentID = STUDENT.StudentID
WHERE ClassID = "CS1" // WHERE (ClassID = "CS1");
```

9 (a) (i) **One mark** for the contents of the accumulator and **one mark** for the reason. **[2]**

Accumulator contents: 0100 0101

Reason:

Address is 60
Contents of the index register is 8
And 60 + 8 = 68 in denary gives the address
The contents of which is 0100 0101 in binary.

(ii) **0000 0111** **[1]**

(b)

Instruction address	Working space	ACC	Memory address				IX	OUTPUT
			100	101	102	103		
			20	100	1	0	1	
50		20						
51		21						
52						21		
53		100						
54		120						
55								
56								
57								
59							2	
60		20						
61		120						
62								'x'
63								

One mark for each shaded block. **[7]**

- Contents of the Accumulator in first 2 lines (instruction addresses 50 and 51)
- Updating address 103 (instruction 52)
- Loading the Accumulator and addition (instructions 53 and 54)
- Not executing instruction 58
- Incrementing the index register (instruction 59)
- Loading the Accumulator and addition (instructions 60 and 61)
- Correct output of 'x' (instruction 62)